

Arquitectura Web

Entregable 3 - 2023

Grupo 10

Christensen, Gabriela

Garrido, Magalí

Iglesias Coqueti, Joaquín

El proyecto está configurado para conectarse a MySQL en localhost puerto 3306.
La API correrá por defecto en localhost:8080

ENDPOINT /estudiantes

GET /estudiantes[?orden=:campo&genero=:genero]

GET

localhost:8080/estudiantes

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettings

Query Params

Key	Value
-----	-------

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON

```
1  [
2    {
3      "lu": 12345,
4      "nombres": "Lucas",
5      "apellido": "Gómez",
6      "edad": 28,
7      "genero": "M",
8      "dni": "54321",
9      "ciudad": "Buenos Aires"
10   },
11   {
12     "lu": 12346,
13     "nombres": "Felipe",
14     "apellido": "Ortega",
15     "edad": 24,
16     "genero": "M",
17     "dni": "54322",
18     "ciudad": "Bahía Blanca"
19   },
20   {
21     "lu": 23456,
22     "nombres": "Sofía",
23     "apellido": "Gutiérrez",
24     "edad": 34,
25     "genero": "F",
26     "dni": "87654",
27     "ciudad": "Salta"
28   },
29   {
30     "lu": 23457,
31     "nombres": "Nicolás",
32     "apellido": "Fernández",
33     "edad": 31,
34     "genero": "M",
35     "dni": "65433",
```

GET /estudiantes/:ID

GET

localhost:8080/estudiantes/34568

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsS

Query Params

	Key
--	-----

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON

```
1  {
2    "lu": 34568,
3    "nombres": "Sebastián",
4    "apellido": "Luna",
5    "edad": 33,
6    "genero": "M",
7    "dni": "76544",
8    "ciudad": "San Luis"
9  }
```

POST /estudiantes

BODY JSON example:

```
{  
  "lu": 311,  
  "nombres": "Tucu",  
  "apellido": "Melo",  
  "edad": 25,  
  "genero": "M",  
  "dni": "12121212",  
  "ciudad": "Tandil"  
}
```

The screenshot displays a REST client interface with the following components:

- Request Bar:** Method **POST** and URL **localhost:8080/estudiantes**.
- Tabs:** Params, Authorization, Headers (9), **Body** (selected), Pre-request Script, Tests, Settings.
- Body Type:** A row of radio buttons for **none**, **form-data**, **x-www-form-urlencoded**, **raw** (selected), **binary**, **GraphQL**, and **JSON** (with a dropdown arrow).
- Request Body Editor:** A text area showing the JSON body:

```
1 {  
2   "lu": 311,  
3   "nombres": "Tucu",  
4   "apellido": "Melo",  
5   "edad": 25,  
6   "genero": "M",  
7   "dni": "12121212",  
8   "ciudad": "Tandil"  
9 }
```
- Response Section:** Tabs for **Body** (selected), Cookies, Headers (5), and Test Results.
- Response Body Editor:** Buttons for **Pretty** (selected), **Raw**, **Preview**, and **Visualize**. A **JSON** dropdown and a refresh icon are also present.
- Response Body:** A text area showing the JSON body in a compact format:

```
1 {  
2   "lu": 311,  
3   "nombres": "Tucu",  
4   "apellido": "Melo",  
5   "edad": 25,  
6   "genero": "M",  
7   "dni": "12121212",  
8   "ciudad": "Tandil"  
9 }
```

PUT /estudiantes/:ID

BODY JSON example:

```
{  
  "nombres": "Santiago",  
  "apellido": "Garcia",  
  "edad": 25,  
  "genero": "M",  
  "dni": "11125",  
  "ciudad": "Buenos Aires"  
}
```

The screenshot shows a REST client interface with the following components:

- Request Bar:** Method **PUT** and URL **http://localhost:8080/estudiantes/12345**.
- Request Headers:** Params, Authorization, Headers (9), **Body** (selected), Pre-reqs.
- Body Type:** none, form-data, x-www-form-urlencoded, **raw** (selected).
- Request Body (Raw):**

```
1 {  
2   "nombres": "Martin",  
3   "apellido": "Garcia",  
4   "edad": 25,  
5   "genero": "M",  
6   "dni": "11125",  
7   "ciudad": "Buenos Aires"  
8 }  
9
```
- Response Section:** Body (selected), Cookies, Headers (5), Test Results.
- Response Body (Pretty):**

```
1 {  
2   "lu": 12345,  
3   "nombres": "Santiago",  
4   "apellido": "Garcia",  
5   "edad": 25,  
6   "genero": "M",  
7   "dni": "11125",  
8   "ciudad": "Buenos Aires"  
9 }
```

DELETE localhost:8080/estudiantes/:ID

DELETE

localhost:8080/estudiantes/34568

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

Query Params

	Key	Value
	Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

true

ENDPOINT /carreras

GET /carreras

GET

localhost:8080/carreras

ParamsAuthorizationHeaders (7)BodyPre-request Script

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "idCarrera": 5,
4     "nombre": "Física"
5   },
6   {
7     "idCarrera": 1,
8     "nombre": "Historia"
9   },
10  {
11    "idCarrera": 4,
12    "nombre": "Matemática"
13  },
14 ]
```

GET /carreras/:ID

GET

localhost:8080/carreras/2

ParamsAuthorizationHeaders (7)BodyPre-rec

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON

```
1 {
2   "idCarrera": 2,
3   "nombre": "TUDAI"
4 }
```

POST /carreras/:ID

BODY JSON example:

```
{  
  "nombre": "Carrera Nueva"  
}
```

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** localhost:8080/carreras
- Body Type:** raw (selected)
- Body Content:**

```
1 {  
2   "nombre": "Carrera Nueva"  
3 }
```
- Body Tab:** Pretty (selected)
- Body Content (Pretty):**

```
1 {  
2   "idCarrera": 10,  
3   "nombre": "Carrera Nueva"  
4 }
```

PUT /carreras/:ID

BODY JSON example:

```
{  
  "nombre": "Veterinaria"  
}
```

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://localhost:8080/carreras/1
- Body Tab:** Body (selected)
- Body Content (Pretty):**

```
1 {  
2   "idCarrera": 1,  
3   "nombre": "Veterinaria"  
4 }
```


DELETE /carreras/:ID

DELETE



localhost:8080/carreras/10

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

Body

Cookies

Headers (5)

Test Results



Status: 200 OK

Pretty

Raw

Preview

Visualize

JSON



1 true

ENDPOINT /inscripciones

GET /inscripciones

GET

localhost:8080/inscripciones

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Body

Cookies

Headers (5)

Test Results

🌐

Status: 200 OK

Time: 70 ms

Size

Pretty

Raw

Preview

Visualize

JSON

🔗

```
1  {
2    "fechaIngreso": "2021-12-07T12:13:37.595+00:00",
3    "fechaEgreso": "2023-08-01T15:25:37.125+00:00",
4    "esGraduado": true,
5    "lu": 67891,
6    "idCarrera": 1,
7    "carrera": "Historia",
8    "estudiante": "Silva, Victoria"
9  },
10 {
11   "fechaIngreso": "2019-03-08T13:35:18.458+00:00",
12   "fechaEgreso": "2023-06-10T21:45:17.199+00:00",
13   "esGraduado": true,
14   "lu": 23458,
15   "idCarrera": 2,
16   "carrera": "TUDAI",
17   "estudiante": "García, Daniel"
18 },
19 {
20   "fechaIngreso": "2023-07-12T21:50:05.002+00:00",
21   "fechaEgreso": null,
22   "esGraduado": false,
23   "lu": 56790,
24   "idCarrera": 2,
25   "carrera": "TUDAI",
26   "estudiante": "Rodríguez, Olivia"
27 },
28 }
```

GET /inscripciones/:LU_ESTUDIANTE/:ID_CARRERA

GET

localhost:8080/inscripciones/67891/1

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	Key	Value
	Key	Value

Body

Cookies

Headers (5)

Test Results

⌐ Status:

Pretty

Raw

Preview

Visualize

JSON

⌵

⌵

1

2

3

4

5

6

7

8

9

{

"fechaIngreso": "2021-12-07T12:13:37.595+00:00",

"fechaEgreso": "2023-08-01T15:25:37.125+00:00",

"esGraduado": true,

"lu": 67891,

"idCarrera": 1,

"carrera": "Historia",

"estudiante": "Silva, Victoria"

}

POST /inscripciones

BODY JSON example:

```
{  
  "fechaIngreso": "2023-10-03T03:15:16.669+00:00",  
  "fechaEgreso": "2025-10-01T16:14:58.000+00:00",  
  "esGraduado": true,  
  "lu": 12345,  
  "idCarrera": 6  
}
```

Sólo lu y idCarrera son obligatorios.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** localhost:8080/inscripciones
- Body Type:** raw
- Request Body (JSON):**

```
{  
  "lu": 12345,  
  "idCarrera": 3  
}
```
- Response Body (JSON):**

```
{  
  "fechaIngreso": "2023-10-09T20:38:51.320+00:00",  
  "fechaEgreso": null,  
  "esGraduado": false,  
  "lu": 12345,  
  "idCarrera": 3,  
  "carrera": "Sistemas",  
  "estudiante": "Gómez, Lucas"  
}
```

PUT /inscripciones/:LU_ESTUDIANTE/:ID_CARRERA

BODY JSON example:

```
{
  "fechaIngreso": "2020-11-07T12:13:37.595+00:00",
  "fechaEgreso": "2023-08-01T15:25:37.125+00:00",
  "esGraduado": true,
  "carrera": "Historia",
  "estudiante": "Silva, Victoria"
}
```

The screenshot shows a REST client interface with the following components:

- Request Bar:** Method **PUT** and URL **http://localhost:8080/inscripciones/67891/1**.
- Request Headers:** Params, Authorization, Headers (9), **Body** (selected), Pre-request Script.
- Body Type:** none, form-data, x-www-form-urlencoded, **raw** (selected), binary.
- Request Body (Raw):**

```
1 {
2   ... "fechaIngreso": "2020-11-07T12:13:37.595+00:00",
3   ... "fechaEgreso": "2023-08-01T15:25:37.125+00:00",
4   ... "esGraduado": true,
5   ... "carrera": "Historia",
6   ... "estudiante": "Silva, Victoria"
7 }
```
- Response Section:** Body (selected), Cookies, Headers (5), Test Results.
- Response Body (Pretty):**

```
1 {
2   "fechaIngreso": "2020-11-07T12:13:37.595+00:00",
3   "fechaEgreso": "2023-08-01T15:25:37.125+00:00",
4   "esGraduado": true,
5   "carrera": "Historia",
6   "estudiante": "Silva, Victoria",
7   "lu": 67891,
8   "idCarrera": 1
9 }
```


DELETE /inscripciones/:LU_ESTUDIANTE/:ID_CARRERA

DELETE

localhost:8080/inscripciones/23458/2

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Body

Cookies

Headers (5)

Test Results

Sta

Pretty

Raw

Preview

Visualize

JSON

1

true

Implementar las consultas mediante servicios REST para:

a) Dar de alta un estudiante

Ver POST estudiantes

b) Matricular un estudiante en una carrera

Ver POST inscripciones

c) Recuperar todos los estudiantes, y especificar algún criterio de ordenamiento simple.

GET /estudiantes?orden=nombres

GET /estudiantes?orden=apellidos

GET /estudiantes?orden=edad

The screenshot shows a REST client interface with a GET request to `localhost:8080/estudiantes/?orden=edad`. The response is a JSON array of three student objects, sorted by age. The interface includes tabs for Params, Authorization, Headers (7), Body, Pre-request Script, and Tests. The Body tab is selected, showing the JSON response in a Pretty view. The response is a JSON array of three student objects, each with fields: `lu`, `nombres`, `apellido`, `edad`, `genero`, `dni`, and `ciudad`.

```
1 [
2   {
3     "lu": 67890,
4     "nombres": "Martina",
5     "apellido": "Martínez",
6     "edad": 23,
7     "genero": "F",
8     "dni": "98765",
9     "ciudad": "Córdoba"
10  },
11  {
12    "lu": 12346,
13    "nombres": "Felipe",
14    "apellido": "Ortega",
15    "edad": 24,
16    "genero": "M",
17    "dni": "54322",
18    "ciudad": "Bahía Blanca"
19  },
20  {
21    "lu": 0,
22    "nombres": "Tucu",
23    "apellido": "Melo",
24    "edad": 25,
25    "genero": "M",
26    "dni": "12121212",
27    "ciudad": "Tandil"
28  },
29  ]
```


d) Recuperar un estudiante, en base a su número de libreta universitaria.

Ver GET /estudiantes/:lu

e) Recuperar todos los estudiantes, en base a su género.

GET /estudiantes?genero=m


GET /estudiantes?genero=f

The screenshot shows a REST client interface with a GET request to `localhost:8080/estudiantes/?genero=f`. The response is a JSON array of three student objects, displayed in a 'Pretty' format. The status is 200 OK and the time taken is 12 ms.


```
1  {
2    "lu": 23456,
3    "nombres": "Sofía",
4    "apellido": "Gutiérrez",
5    "edad": 34,
6    "genero": "F",
7    "dni": "87654",
8    "ciudad": "Salta"
9  },
10 {
11   "lu": 34567,
12   "nombres": "Isabella",
13   "apellido": "López",
14   "edad": 35,
15   "genero": "F",
16   "dni": "76543",
17   "ciudad": "San Miguel de Tucumán"
18 },
19 {
20  "lu": 45678,
21  "nombres": "Catalina".
```



f) Recuperar las carreras con estudiantes inscriptos, y ordenar por cantidad de inscriptos.

GET /carreras/ordenar-por-inscriptos

GET  localhost:8080/carreras/ordenar-por-inscriptos

Params Authorization Headers (7) Body Pre-request Script Tests

Body Cookies Headers (5) Test Results 

Pretty Raw Preview Visualize JSON  

```
1  [
2    {
3      "idCarrera": 2,
4      "nombre": "TUDAI",
5      "cantidadInscriptos": 4
6    },
7    {
8      "idCarrera": 3,
9      "nombre": "Sistemas",
10     "cantidadInscriptos": 3
11   },
12   {
13     "idCarrera": 1,
14     "nombre": "Historia",
15     "cantidadInscriptos": 2
16   },
17   {
18     "idCarrera": 6,
19     "nombre": "TUARI",
20     "cantidadInscriptos": 1
21   }
22 ]
```

g) Recuperar los estudiantes de una determinada carrera, filtrado por ciudad de residencia.

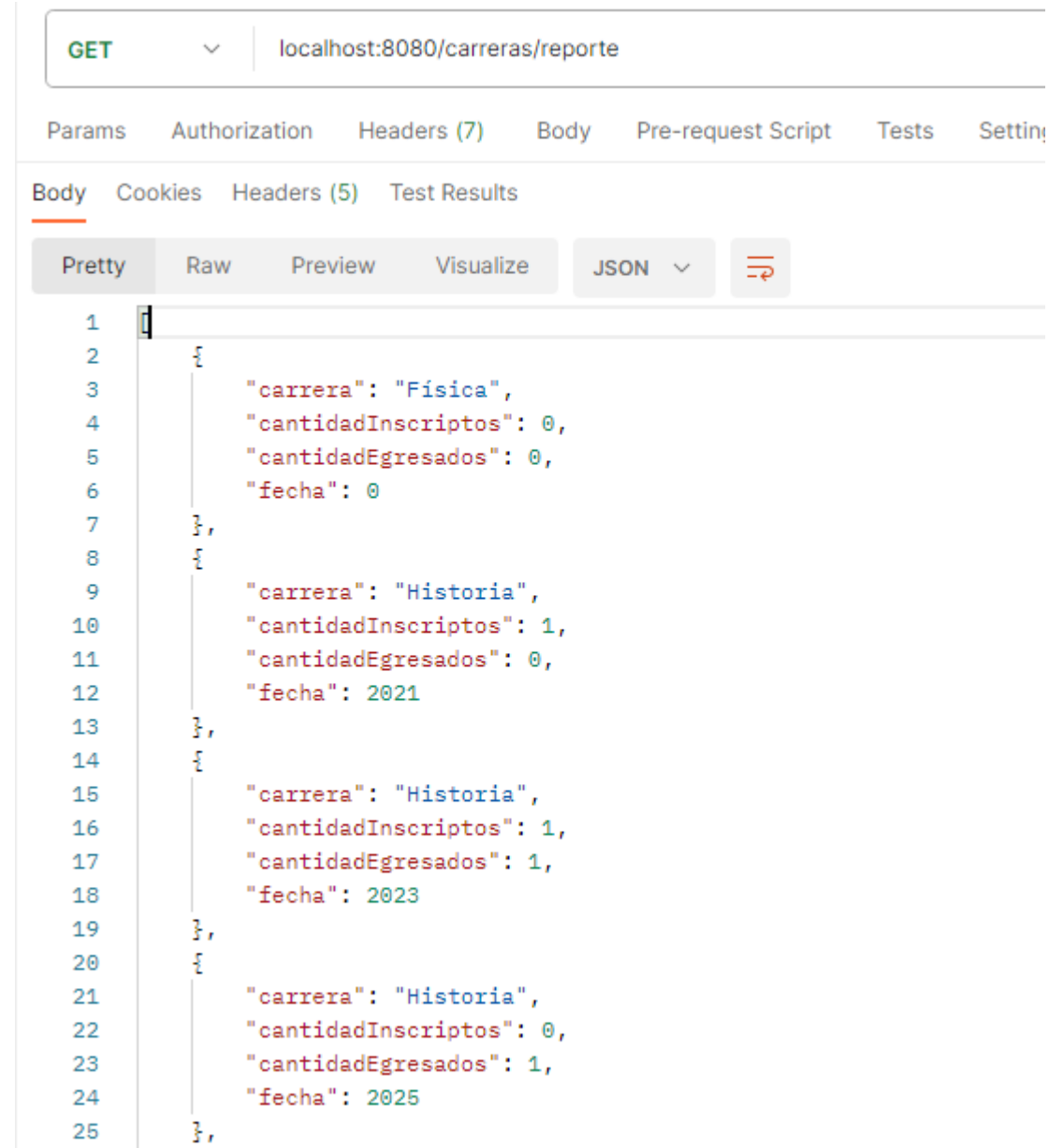
GET /estudiantes/carrera-ciudad?carrera=1&ciudad=neuquen

The screenshot shows a REST client interface. At the top, a dropdown menu is set to 'GET' and the URL 'localhost:8080/estudiantes/carrera-ciudad?carrera=1&ciudad=neuquen' is entered. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is selected, and within it, there are sub-tabs for 'Body', 'Cookies', 'Headers (5)', and 'Test Results'. The 'Body' sub-tab is active, showing a JSON response in 'Pretty' format. The JSON object contains the following fields: 'lu' (67891), 'nombres' ('Victoria'), 'apellido' ('Silva'), 'edad' (28), 'genero' ('F'), 'dni' ('98766'), and 'ciudad' ('Neuquén').

```
1 [
2   {
3     "lu": 67891,
4     "nombres": "Victoria",
5     "apellido": "Silva",
6     "edad": 28,
7     "genero": "F",
8     "dni": "98766",
9     "ciudad": "Neuquén"
10  }
11 ]
```

h) Generar un reporte de las carreras, que para cada carrera incluya información de los inscriptos y egresados por año. Se deben ordenar las carreras alfabéticamente, y presentar los años de manera cronológica

GET /carreras/reporte



The screenshot shows a REST client interface with a GET request to `localhost:8080/carreras/reporte`. The response body is displayed in JSON format, showing an array of three objects. Each object represents a career with its name, the number of registrants, the number of graduates, and a year. The careers are ordered alphabetically: Física, Historia, and Historia. The years are 0, 2021, and 2023 respectively.

```
1  [
2    {
3      "carrera": "Física",
4      "cantidadInscriptos": 0,
5      "cantidadEgresados": 0,
6      "fecha": 0
7    },
8    {
9      "carrera": "Historia",
10     "cantidadInscriptos": 1,
11     "cantidadEgresados": 0,
12     "fecha": 2021
13   },
14   {
15     "carrera": "Historia",
16     "cantidadInscriptos": 1,
17     "cantidadEgresados": 1,
18     "fecha": 2023
19   },
20   {
21     "carrera": "Historia",
22     "cantidadInscriptos": 0,
23     "cantidadEgresados": 1,
24     "fecha": 2025
25   },
26 ]
```